NISTIR 6042

The Role of Product Data Management in the Manufacturing Engineering ToolKit

Michael Iuliano

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards and Technology
Gaithersburg, MD 20899-0001

NIST

NISTIR 6042

The Role of Product Data Management in the Manufacturing Engineering ToolKit

Michael Iuliano

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards and Technology
Gaithersburg, MD 20899-0001

August 1997

NIST IR 6042

The Role of Product Data Management in the Manufacturing Engineering ToolKit

Michael J. Iuliano
Manufacturing Systems Integration Division
Manufacturing Engineering Lab
National Institute of Standards and Technology, Gaithersburg MD, USA

Abstract

A Manufacturing Engineering ToolKit (METK) is under development at the National Institute of Standards and Technology (NIST). The toolkit consists of commercial-off-the-shelf (COTS) manufacturing engineering software applications. The purpose of the toolkit is to provide an integrated framework, operating environment, common databases, and interface standards for those applications. Currently, manufacturing software applications provide computer-aided support for most of the engineering activities required to plan the processes needed to make parts. The data generated in each of these activities must be captured and archived. Since most of the data is in the form of computer data files, a configuration management functionality is needed to capture/maintain the data files and make the files available to the various engineering applications. A Product Data Management (PDM) application provides this functionality. PDM applications allow the capture, archival, and recovery of data files with revision and version control. This paper describes how a PDM application is being used in the Manufacturing Engineering ToolKit.

1 Introduction

The Manufacturing Engineering ToolKit (METK) [1] is under development at the National Institute of Standards and Technology (NIST). The toolkit consists of commercial-off-the-shelf (COTS) software applications. These applications support various engineering activities in a manufacturing enterprise. The METK focus is the following manufacturing engineering functions: design, process planning, and engineering data validation. The purpose of the METK at NIST is to provide an integrated framework, operating environment, common databases, and interface standards for manufacturing engineering software applications. The goal is to lower manufacturing costs, reduce delivery times, and improve product quality through the use of advanced and integrated software tools.

The Manufacturing Engineering ToolKit currently consists of the following software:
1) Adra Systems' Matrix ™ Version 3.0 is a product data management application,
2) Parametric Technology Corporation's Pro-Engineer ™ is a CAD application,

3) Technomatix's ICEM ™ Part is a generative process planning application, and
4) Deneb Robotics Igrip ™, Quest ™ and Virtual NC ™ are manufacturing simulation applications used for data validation.

The PDM application is being used in the Manufacturing Engineering ToolKit to capture the engineering data generated by a toolkit application. After the data is captured, it is then made available to other toolkit applications.

The METK project is being conducted as a collaborative effort between software developers, manufacturing engineers and managers, and researchers from industry, vendors, government and academia. Through consensus, the METK's focus has been on the validation of manufacturing engineering data. The idea of the METK is to use computer simulation models to validate data before the data is physically used to generate products [1] and capture the data using a PDM application (See Figure 1).

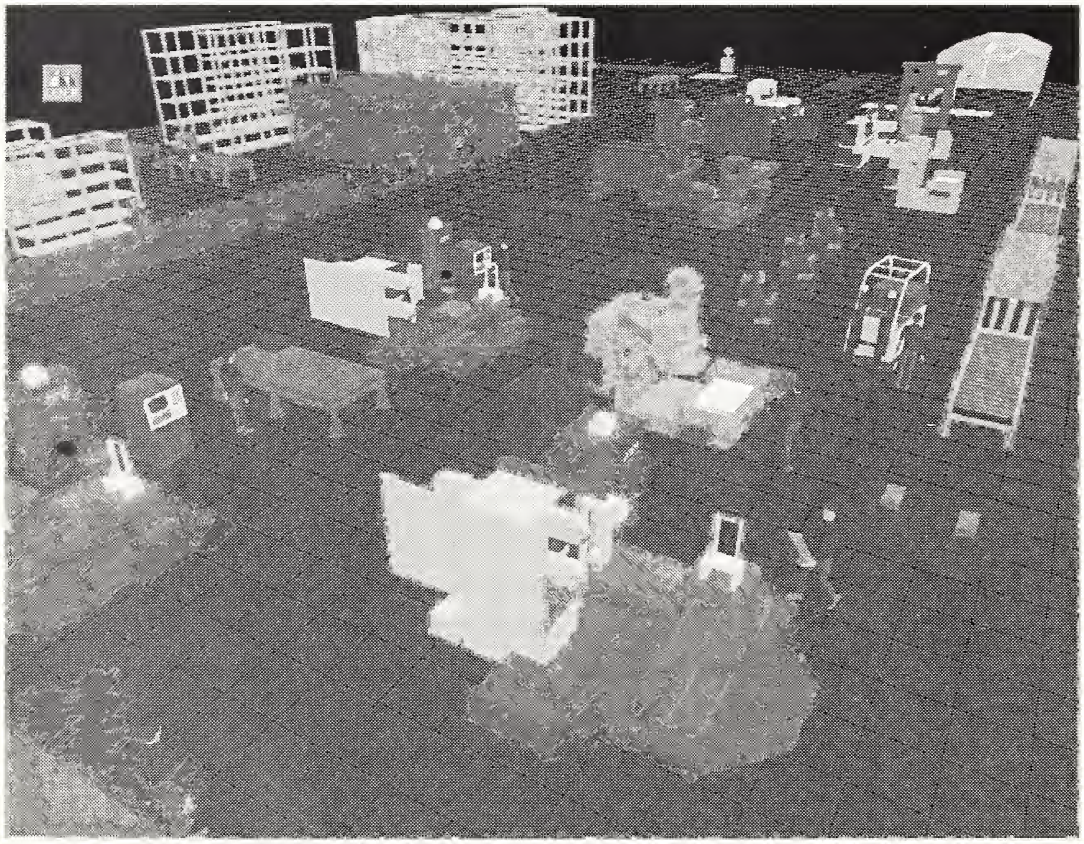


Figure 1: Using computer applications to generate, capture and validate data.

Section 2 of this paper describes the information management problem that is facing manufacturers today and how Product Data Management applications provide a solution for some of the information management requirements. Section 3 describes how a Product Data Management application is being used in the METK and includes screen captures from the application that describe the implementation within the application which supports the METK. Section 4 is a summary and presents conclusions about the usage of a Product Data Management application in the METK.

2 The Information Management Problem

Manufacturers are increasingly using computers to perform activities within their organizations. As the amount of data generated by computer applications continues to grow, so do the requirements for managing the data. This data consists of such things as CAD drawings, numerical control tape images, process sheets, text files, spreadsheet data, computer video, simulation models, analysis results and graphics files.

Further complicating the problem is that the information management requirements for an individual manufacturer depends on a variety of factors. Some factors that might influence the informational management requirements are the software applications, operating systems, databases, employee skill sets in place at an organization. The type of product being made may also influence the requirements. All things considered, information management is a difficult task and be very problematic.

In modern day manufacturing enterprises, the information management requirements are in mainly in the electronic domain. Since most of the data is in the form of computer data files, a configuration management function is needed that allows the capture, archival, and recovery of data files with revision and version control. Version control ensures all changes to data are stored with a history. Revision control ensures the proper versions of component data elements are combined in a meaningful way.

In addition, data must be maintained with proper access control, integrity, backup, and recovery procedures. Access control ensures no data is accessible without proper authorization. Access control eliminates problems such as the accidental use of incorrect product data or security violations. These types of problems lead to lost productivity and lower competitiveness. The levels generally range from file data encryption in protected server directories to the file data remaining in user file system directories. Integrity validates all data is complete and of the correct format. Backup and recovery allow the restoration of data to any state in the past. This is useful if something corrupts any of the data such as a human error or a system failure.

2.1 Product Data Management (PDM) applications

To address these very complex information management requirements, a new type of computer application is being used, Product Data Management (PDM) applications. Commercial PDM software applications are being installed in a growing number of manufacturing companies. The worldwide market for PDM systems grew 36% between 1993 and 1994. According to leading PDM consulting firm [3], through to the end of the decade, product data management applications will be one of the fastest-growing segments of the software marketplace. The market for PDM software and services presently exceeds $225 million per year. Given its current compounded annual growth rate of 40 percent, PDM will be a billion-dollar industry within five years.

Along with the rise in PDM application popularity came the rise in the number of PDM application vendors. Since it is the nature of software that a computing problem can be solved with a variety software solutions, each of the PDM vendors developed and offered their own software solution. These software solutions shared some common ideas but differed in the priorities and implementation. The result is that each PDM application has it's own strong points and capabilities. These applications must continue to evolve so they can be fully scaleable across different types of manufacturers.

2.2 Current PDM technology and future PDM requirements

Current PDM applications offer a wide range of functions, including engineering information management, engineering change management, product structure management, product configuration control, workflow management, and document management. PDM applications are currently being applied throughout the entire product life cycle.

PDM applications must be developed with an open architecture. This will allow organizations to integrate their current manufacturing engineering support applications into a PDM application with minimal difficulty. This also would help eliminate time-consuming transfer of data and training which are inherent when switching to a new software technology.

PDM applications must be developed to run in a heterogeneous computing environment. Modern manufacturing organizations have different computing environments in different locations within the company. Most of the time these environments use different hardware platforms and operating systems. PDM applications must be able to handle data across all platforms to avoid engineers not having access to the proper data.

PDM applications should accommodate product structure. For a product, the relationship between its component assemblies and between the parts that make up these assemblies must be maintained. This would allow engineers to open a complete Bill of Materials, including documents and parts, either for the entire product or selected assemblies. One clear requirement for a PDM application is the ability to hold not just the physical relationships between parts in an assembly but also other kinds of structures or relationships; for instance, manufacturing, financial, maintenance or document relationships.

A PDM application should act as the engineer's working environment, meticulously capturing all new and changed data as it is generated, maintaining a record of which version it is, recalling it on demand and effectively keeping track of the engineer's every move.

A PDM application should provide organizational support. This will allow the modeling of individuals, the many groups they may work in, and the various roles they may have over time. This fits well in today's economy where companies are continuously down sizing and utilizing reduced personnel staffs in many different roles.

3 Using a PDM application in the Manufacturing Engineering ToolKit

The METK project is centered around the concept of an engineering data package. The engineering data package is thought of as a package containing all the engineering data elements required for the production of a part. In the METK, the engineering data elements consist of tool lists, fixture lists, nc programs, operations sheets, route sheets, and geometry models. By restricting the project's scope to these elements, the project can focus the integration of computer-aided manufacturing engineering applications and the validation of the engineering data [2].

A geometry model is the CAD models for a product design. A route sheet specifies a sequence of workstations which a workpiece must visit. An operations sheet contains a set of sequenced machining operations to be performed at a specific workstation with a specific workpiece setup. A tool list is a list of tools required to machine the part at the workstation. A fixture list is a list of fixtures required for a specific workpiece setup. A NC program is a set of instructions indicating machine movements required to machine the workpiece. A NC program is machine specific and is typically prepared for a single workpiece setup. All of these engineering data package elements exist in the form of electronic data files. See Figure 2 for a overview of the METK system.

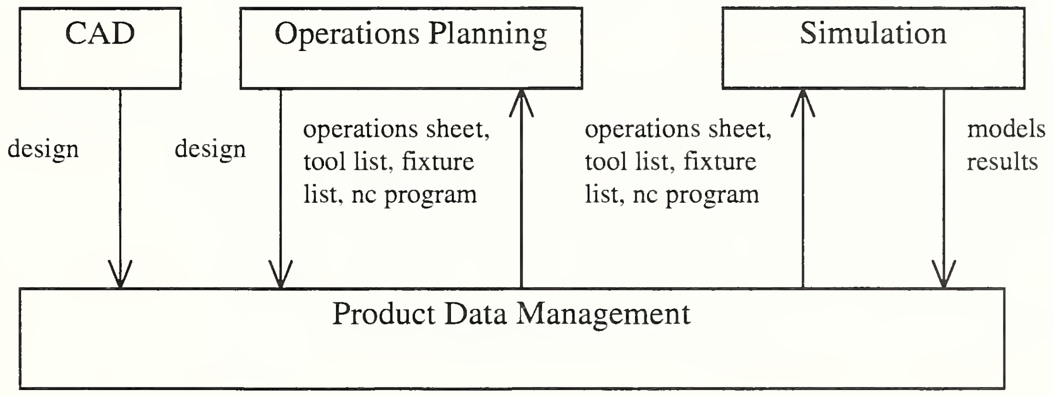
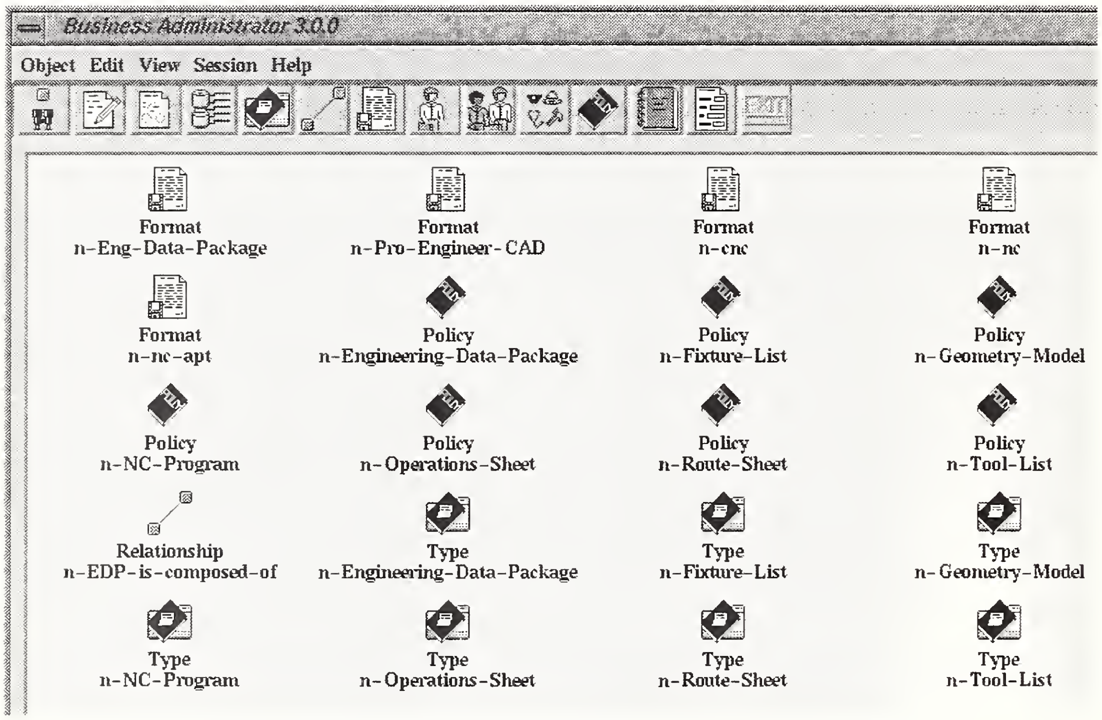

CAD
Operations Planning
Simulation
design
design
operations sheet, tool list, fixture list, nc program
operations sheet, tool list, fixture list, nc program
models results
Product Data Management

Figure 2 : METK system overview.

3.1 METK Product Data Management definitions

This section describes how we interacted with the PDM application to implement PDM entities to support the METK project. Within the PDM application, definition types were created to support the engineering data package concept. These definitions consist of types, policies, formats, and relationships. The type definition types correspond to each of the data elements in the engineering data package In this section, actual screen captures from the PDM application will be shown to illustrate the METK implementation. Explanation will be given to describe the implementation.

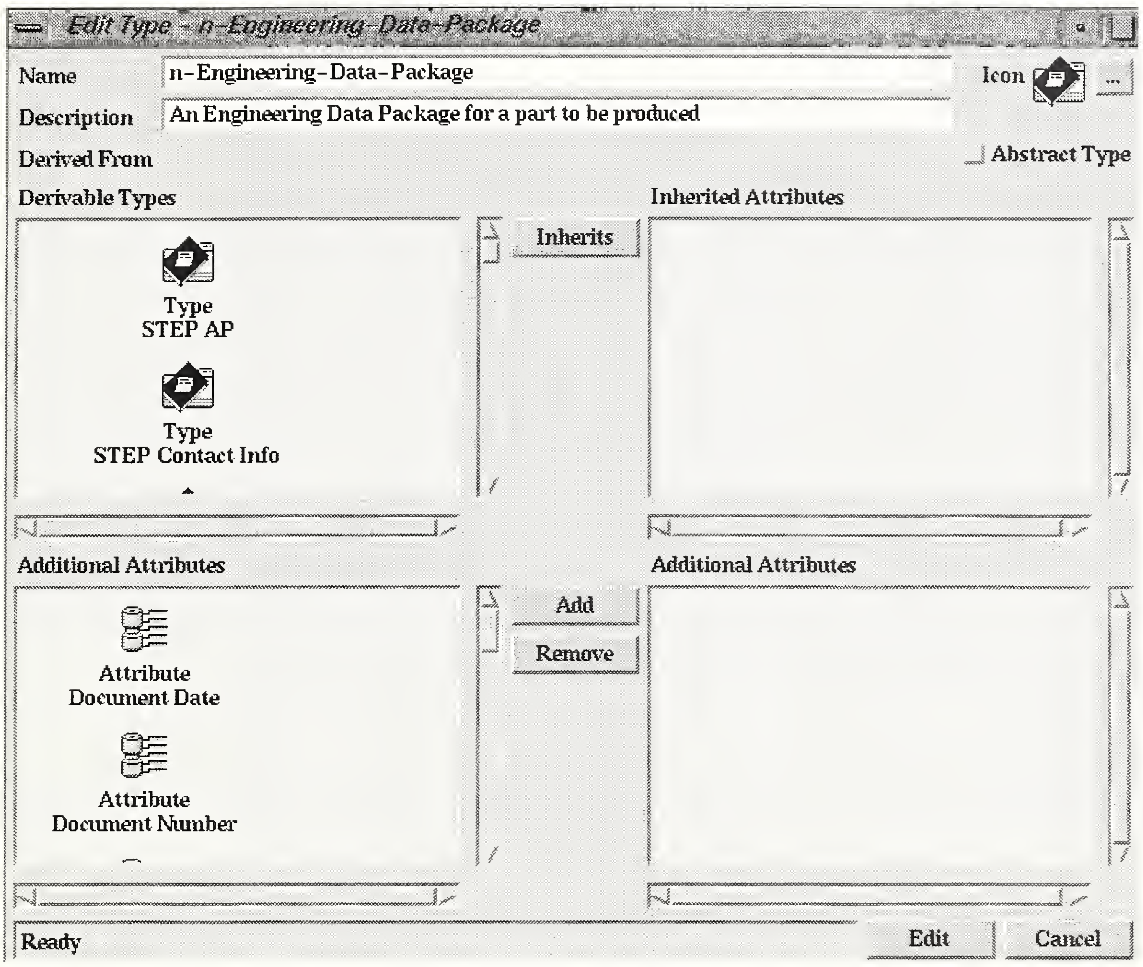


Format n-Eng-Data-Package
Format n-Pro-Engineer-CAD
Format n-cnc
Format n-nc
Format n-nc-apt
Policy n-Engineering-Data-Package
Policy n-Fixture-List
Policy n-Geometry-Model
Policy n-NC-Program
Policy n-Operations-Sheet
Policy n-Route-Sheet
Policy n-Tool-List
Relationship n-EDP-is-composed-of
Type n-Engineering-Data-Package
Type n-Fixture-List
Type n-Geometry-Model
Type n-NC-Program
Type n-Operations-Sheet
Type n-Route-Sheet
Type n-Tool-List

Figure 3: PDM administrator window with METK definition types

Figure 3 shows the main PDM administrator window with the METK definitions types. The definitions that appear in Figure 3 will be described below. Other possible PDM definition types include attributes, groups, roles and people. PDM definition types define users, business objects, and the characteristics and controls associated with them. Each definition describes an element of a business with a collection of values.

The type definition for the engineering data package is shown in Figure 4.



Edit Type - n-Engineering-Data-Package

Name: n-Engineering-Data-Package

Description: An Engineering Data Package for a part to be produced

Derived From

Abstract Type

Derivable Types: Type STEP AP; Type STEP Contact Info

Inherited Attributes

Additional Attributes: Attribute Document Date; Attribute Document Number

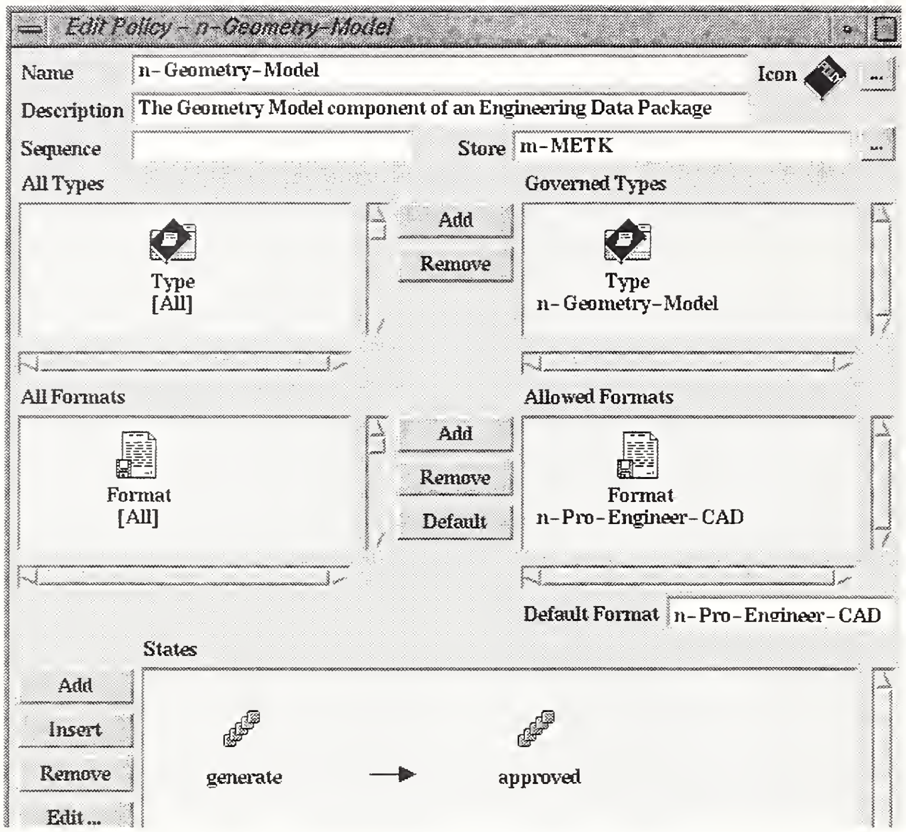
Additional Attributes

Ready

Figure 4: Engineering Data Package type definition.

A type definition describes a business object. In Figure 4, a name and description is given to a type definition. In the box named ‘Derivable Types’ in the figure is a list of existing types. This allows types to be derived from other types. In the boxed named ‘Additional Attributes’ in the figure is a listing of any additional defined attributes that can be added to the type. Attributes define a characteristic of the type. It is the type definition that is instantiated to create a specific PDM object [4]. Objects in the PDM applications are instantiations of type definitions. When the objects are created in the PDM, they are containers for data files. This type definition is a template from which many unique instances can be created.

In the PDM application, types are governed by a policy. Figure 5 shows the policy definition for the geometry model component of the engineering data package.


Edit Policy - n-Geometry-Model

Name: n-Geometry-Model

Description: The Geometry Model component of an Engineering Data Package

Sequence:

Store: m-METK

All Types: Type [All]

Governed Types: Type n-Geometry-Model

All Formats: Format [All]

Allowed Formats: Format n-Pro-Engineer-CAD

Default Format: n-Pro-Engineer-CAD

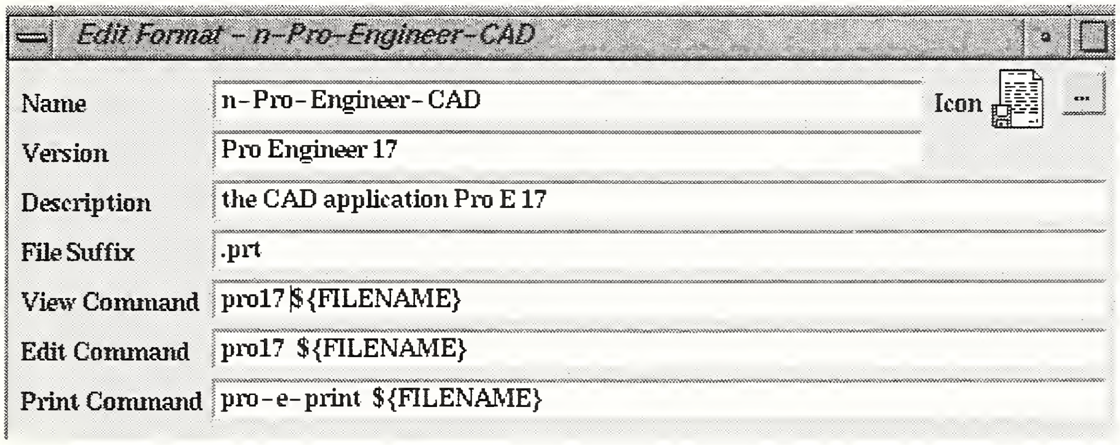
States: generate → approved

Figure 5: Geometry model policy definition.

A policy definition controls a PDM object. It specifies many characteristics about the object including the rules that govern access, approvals, life cycle and revisioning. The information defined in the policy includes: the types of objects the policy will govern, the types of formats allowed for objects, the default format automatically assigned to objects and how revisions will be labeled. The policy dictates the life cycle of the object. A life cycle consists of a series of connected states, each of which represents the stage in the life of the governed object. The purpose of the state is to define: the current state of the object, who will have access to the object, the type of access allowed, whether or not the object can be revised, the conditions required for changing the state of the object.

When states are defined and added to a policy, several additional characteristics of the object can be specified including: notification can specify a message to be sent to indicated users that the object has entered the state being defined; an ownership routing that automatically reassigns the object to another person when the object reaches the state being defined; an operating system command that is executed when the object arrives in the state being defined; and signature requirements that specifies who can control the promotion of an object through it's life cycle.

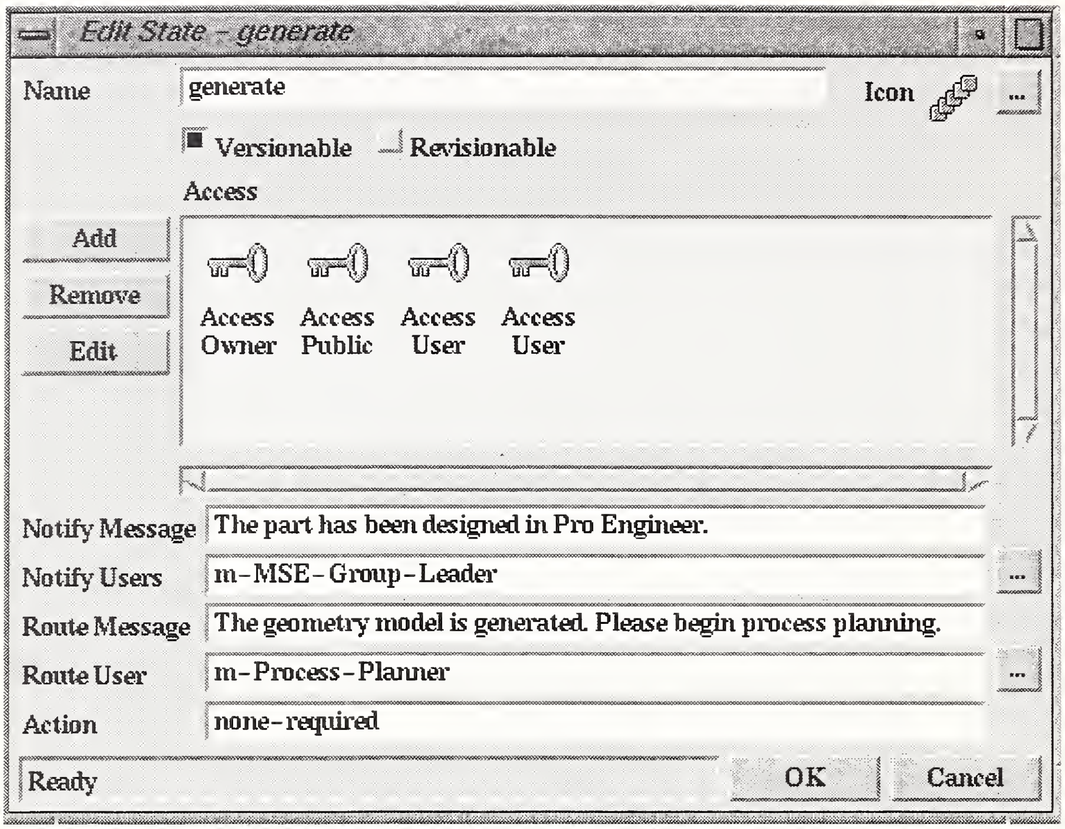
In the boxed named 'Allowed Formats' in figure 5, the default format and the only allowed format of the geometry model is n-Pro-Engineer-CAD. The definition shows the geometry model object will have a life cycle of two states: generate and approved.



Name	n-Pro-Engineer-CAD
Version	Pro Engineer 17
Description	the CAD application Pro E 17
File Suffix	.prt
View Command	pro17 ${FILENAME}
Edit Command	pro17 ${FILENAME}
Print Command	pro-e-print ${FILENAME}

Figure 6: Format definition for n-Pro-Engineer-CAD.

Figure 6 shows the format definition for n-Pro-Engineer-Cad. A format definition in the PDM is used to capture information about different application file formats. A format stores the name of the application, the application version, and the suffix (file extension) used to identify files. It also contains the commands necessary to automatically launch applications to view, edit and print data files of the format. A business object can have many file formats and they are linked to the appropriate type definition by the policy definition.



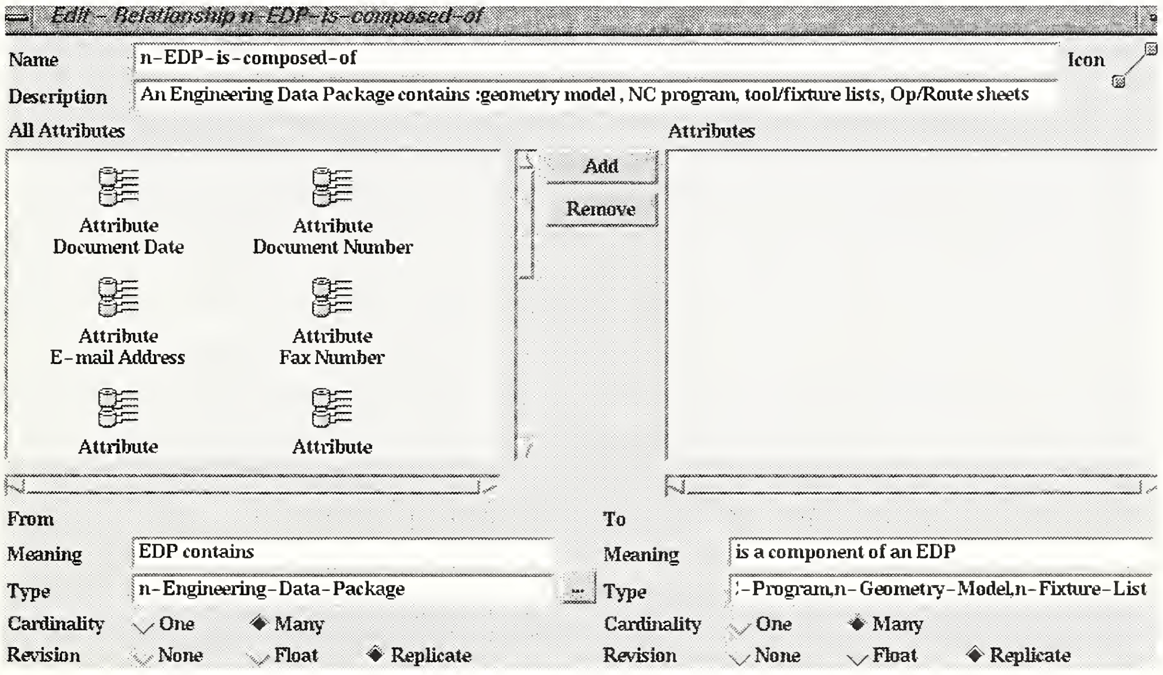
Name: generate
[✓ Versionable] Revisionable
Access
Notify Message: The part has been designed in Pro Engineer.
Notify Users: m-MSE-Group-Leader
Route Message: The geometry model is generated. Please begin process planning.
Route User: m-Process-Planner
Action: none-required
Ready

Figure 7: State definition window for the geometry model's generate state.

Figure 7 shows the state definition window for the generate state of the geometry model policy. There are selections for revisioning and versioning. If the revisionable box is selected, the object is allowed to be revised in this state. When an object is revised, duplicate object is made with a new name that includes contained data files. If the versionable box is selected, multiple files are allowed to be checked into the object in this state. If unselected, only one file is allowed to be contained in the object. Access to instances of objects governed by this policy are defined here. An access is defined for each person who will interact with objects governed by the policy. Access types include: modify, delete, checkin, checkout, schedule, lock, unlock, connect, disconnect, create, promote, and demote.

A relationship links two objects. The objects may be of different types or the same type. The types of objects allowed, the meaning of the relationship, and the rules for maintaining the relationship are specified when the relationship is defined. The objects are connected "from" and "to" one another. Cardinality of the relation is also specified. Allowed cardinalities are: one to one, one to many, many to one, and many to many.

Figure 8 shows the definition window for the relationship n-EDP-is-composed-of. This is name is short for an Engineering Data Package (EDP) is composed of. A relationship is a connection between related objects. The PDM allows two relationship variations: equitable or hierarchical. Equitable relationships are useful for creating sets of related objects. The relationship is equal between both object types. This is useful for showing alternatives or general associations. In a hierarchical relationship, there is clearly a directional flow. One object might parent the other. A hierarchical relationship can link: a single child object to a parent object, multiple child objects to a single parent and multiple parent objects to a single child.



Edit - Relationship n-EDP-is-composed-of

Name	n-EDP-is-composed-of
Description	An Engineering Data Package contains :geometry model, NC program, tool/fixture lists, Op/Route sheets

All Attributes: Attribute Document Date; Attribute Document Number; Attribute E-mail Address; Attribute Fax Number; Attribute; Attribute

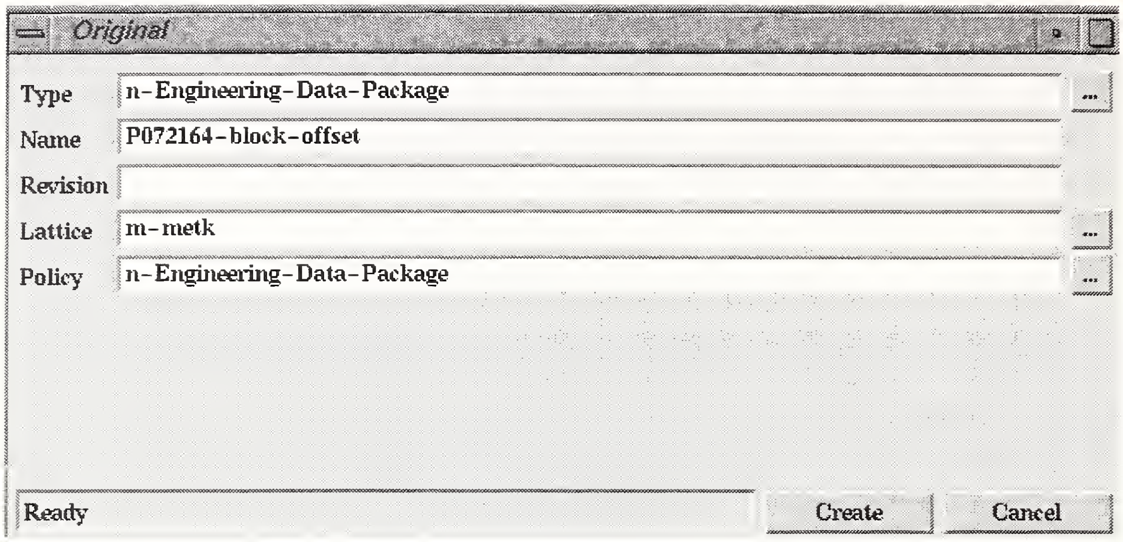
	From	To
Meaning	EDP contains	is a component of an EDP
Type	n-Engineering-Data-Package	:-Program,n-Geometry-Model,n-Fixture-List
Cardinality	One [✓ Many]	One [✓ Many]
Revision	None Float [✓ Replicate]	None Float [✓ Replicate]

Figure 8: Relationship definition window for the Engineering Data Package.

3.2 Creating METK objects

Once all of the METK definition types to support the engineering data package are entered into the PDM application, objects can be created. Objects in the PDM applications are instantiations of type definitions. When the objects are created in the PDM, they are containers for data files.

PDM objects have been created for the METK. The following screen captures show how the definitions are instantiated to create objects and form relationships between objects. Once the objects are created, data files can be generated by appropriate METK applications and can be checked into the container objects in the PDM application.

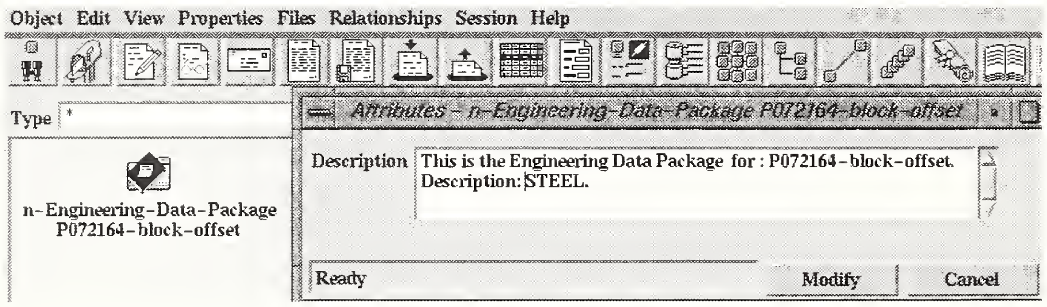


Type	n-Engineering-Data-Package
Name	P072164-block-offset
Revision	
Lattice	m-metk
Policy	n-Engineering-Data-Package

Ready

Figure 9: Creation of an object instance of an Engineering Data Package

Figure 9 shows the creation window for an object instance of the n-Engineering-Data-Package type. A name is supplied for the object. The lattice is where the data files for the object are to be stored. The policy governing the new object is also specified. When created, any data attributes for the object type are specified. Figure 10 shows the data entry window for the description attribute for the object.


Object Edit View Properties Files Relationships Session Help
Attributes - n-Engineering-Data-Package P072164-block-offset
Modify
Cancel

Type *

n-Engineering-Data-Package P072164-block-offset

Description: This is the Engineering Data Package for : P072164-block-offset. Description: STEEL.

Ready

Figure 10: Attribute data entry window layered on top of main PDM window

All of the other objects for the engineering data package are create in a similar way. Once all the object instances are generated, they can be linked together in meaningful way via a relationship instantiation. Figure 11 shows the create relationships window for the engineering data package.

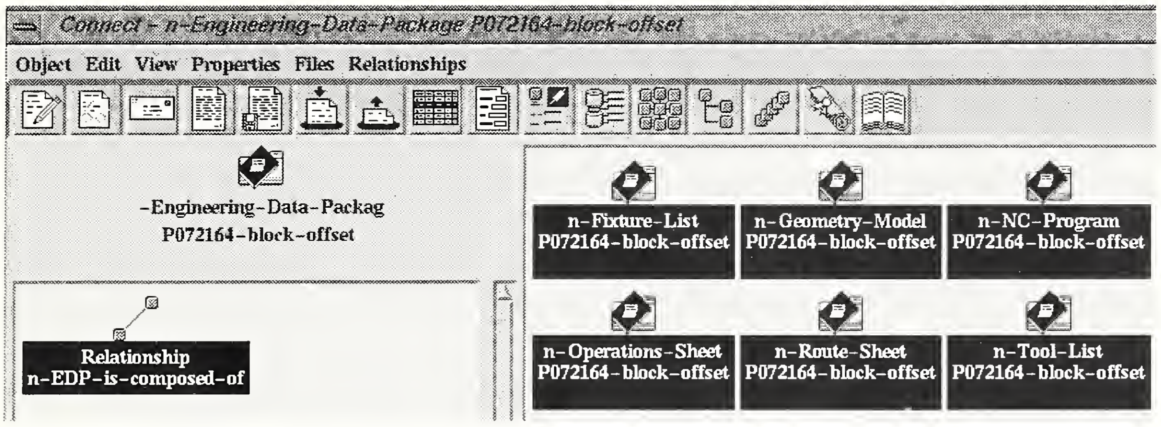


Figure 11: Create relationships window.

In figure 11, the available relation definitions for the n-Engineering-Data-Package type are listed on the left. In this case, only one relationship definition is available: n-EDP-is-composed-of. Once the relationship is created, the objects are linked via the relationship.

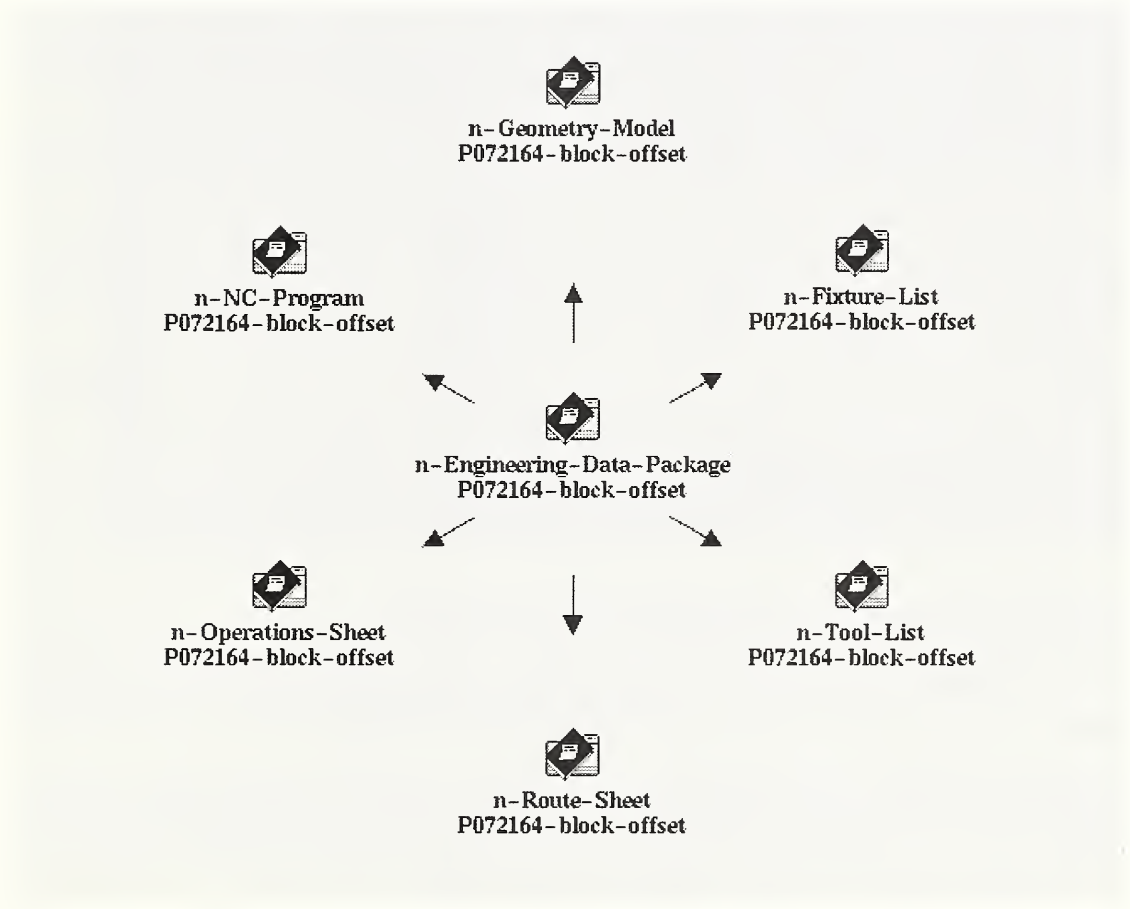


Figure 12: PDM relationship browser showing the engineering data package and related objects.

In figure 12, the PDM relationship browser indicates the object named P072164-block-offset of type n-Engineering-Data-Package is related in a parent/child relationship with the objects that are named the same but of the types of the engineering data package elements.

3.3 Using METK objects.

In the METK, a data file is generated by the CAD application for the product design. This data file is inserted into the PDM geometry model object. Once the data file is generated, the file is checked into the geometry model object in the PDM. The PDM captures the data file. Once this operation is complete, the geometry model object is ready to be promoted to the next state in it's life cycle. Figure 13 shows the promotion window for the geometry model.

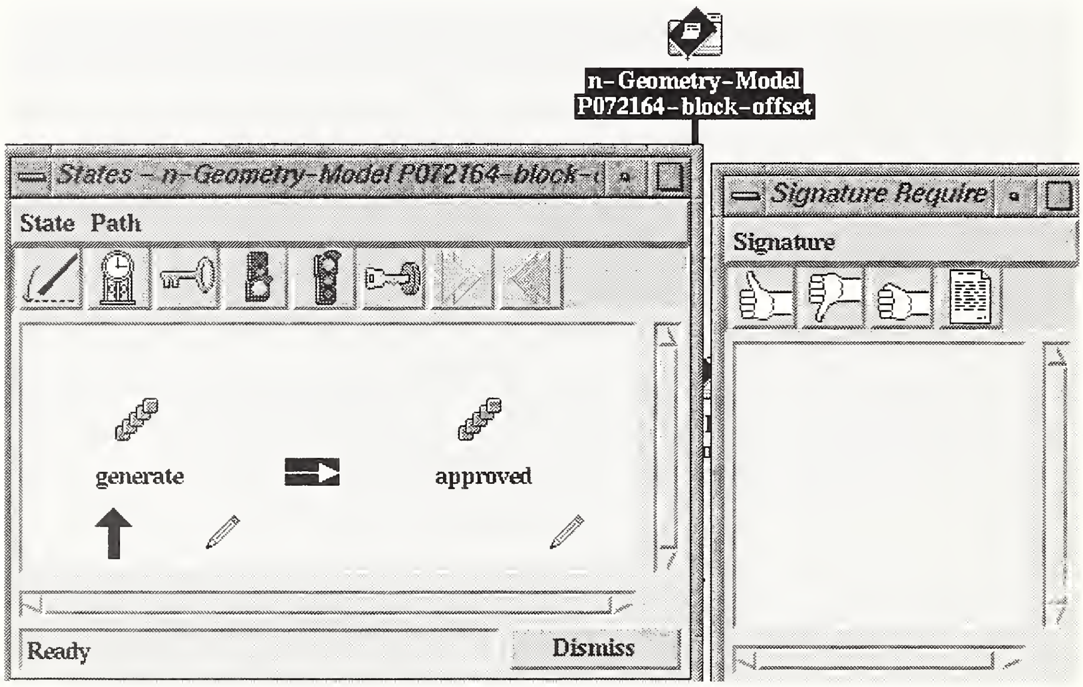


Figure 13: Promotion window for the geometry model object.

Figure 13 indicates the geometry model is currently in the generate state. It also indicates a signature is required to promote the object to the approved state.

All of the METK objects are used in this way in the business flow model. A data file is generated for the object and then checked into the PDM application under the object's control. The object is then promoted through it's life cycle.

4 Summary and Conclusions

A manufacturing Engineering toolkit is under development at NIST. This toolkit consists of Commercial-Off-The-Shelf manufacturing software applications. A Product Data Management (PDM) application is also included in the toolkit. The PDM is being used to capture the data generated by toolkit applications. A business flow model is also implemented in the PDM application to support CAME project objectives. This business model sets the context for METK utilization. This paper described the PDM application being used in the toolkit project and how the business case supporting the project was developed and implemented in the PDM application.

Product Data Management applications are effective in their use to capture, track and validate engineering data. This computer-aided technology will allow manufacturers to capture product data electronically allowing for more efficient tracking, validation, recovery, and retrieval of data. The result will be a shorter time to market for products since manufacturing data will be managed in a much more efficient manner than in the past.

Disclaimer

No approval or endorsement of any commercial product by the National Institute of Standards and Technology is intended or implied.

Acknowledgment

Work described in this paper was sponsored by the U.S. Navy Manufacturing Technology program's Computer-Aided Manufacturing Engineering (CAME) project and the Systems Integration for Manufacturing Applications (SIMA) program being conducted at NIST.

Glossary

Attributes -An attribute is any characteristic that can be assigned to an object or relationship. Objects can have attributes such as size, shape, weight, color, materials, age, texture, etc. [3], [4]

Checkin - Once an object is created, the PDM application allows the user to place files under the control of an object. When a file is placed under control of an object, it is called "checking it into" the PDM system (checkin). [3]

Checkout - Is the term for when you retrieve a file from an object and place a copy of the file on a computer workstation. A master copy of the file is kept in the PDM application. [3]

Instantiate - The term used to describe when a type definition is used as a template for creating objects. An new object instance of the template type is created and any unique data for the instance is entered into the template's attributes. [4]

Policy - A policy controls an object. It specifies *many* aspects about the object including the rules that govern access, approvals, life cycle, and revisioning.

Type - A type defines a kind of object and collection of attributes that characterize it. A type can be derived from another type. A type can be instantiated many times to create many unique instances of objects. [4]

References

[1] Iuliano M, *"Overview of the Manufacturing Engineering ToolKit Prototype"*, NISTIR 5730, October 1995.
[2] *Computer-Aided Manufacturing Forum Second Technical Meeting*, Aug. 22-23, 1995, NISTIR 5846, May 1996.
[3] *Matrix 3.0 Users Guides*, Adra Systems, Inc. Chelmsford, MA, 1995.
[4] Booch, G, *"Object Oriented Analysis and Design"*, Addison Wesly, Second Edition, 1994.